

A Product Line for Workflow Management Systems

Itana Maria de Souza Gimenes
Universidade Estadual de Maringá
Departamento de Informática
itana@din.uem.br

Ruy Tsutomu Nishimura
Universidade Estadual de Maringá
Departamento de Informática
ruy@dc.unifil.br

Edson Alves de Oliveira Junior
Universidade Estadual de Maringá
Departamento de Informática
edson@din.uem.br

Leonor Barroca
The Open University
Department of Computing
l.barroca@open.ac.uk

Fabrcio Ricardo Lazilha
Centro Universitrio de Maringá
Departamento de Informática
fabrcio@cesumar.br

Abstract

This paper presents an ongoing research on the development of a component-based product line architecture for workflow management systems. The research includes a process to design the proposed product line, the population of the product with components, the product member production process, architecture evaluation mechanisms using architecture description languages and support tools, a variability management process and the development of case studies to evaluate each proposed technique.

1. Introduction

A software product line [1] is a collection of systems that share a manageable set of features amongst its main artefacts. These artefacts include a base architecture and a set of common components that populate the architecture. The design of a product family must consider similarities and variabilities amongst its products. Thus, the product line approach is appropriate for domains where there is a demand for specific products that can be modelled from a set of common features and well-defined variability points. WfMS (Workflow Management Systems) support definition, management and execution of workflows [2]. The WfMS domain is appropriate to the application of the product line approach due to both organisation's needs and the efforts of the Workflow Management Coalition (WfMC) to establish a standard architecture and a reference model for them.

This paper presents a component-based product line for WfMS and the research issues dealt with in its development process.

2. The Product Line Architecture Development Process

Product line methods should provide mechanisms to capture and represent domain features and variabilities.

Most of the existing methods are based on domain engineering. In general, they are strong in domain modelling, but less efficient to represent architectures and components. Thus, Component-Based Development (CBD) methods can be used in the design of product lines to bridge the gap between domain analysis and the architecture and component design and implementation. General purpose CBD methods are easy to understand and use. In addition, there are commercial tools that may be used to support them. The proposed product line development process followed the Catalysis method [3] with a UML extended notation to represent variability [4]. Product line approaches that also take CBD into account are Kobra and GenVoca, but they are, as yet, less disseminated than Catalysis.

3. The Research Issues

3.1 Architectural Design and Population

The design of the product line goes from the domain modelling to the product line architecture and internal component development. In the domain modelling, it is already possible to identify similar aspects and variation points amongst product line members. The reference model and generic architecture for WfMS of the WfMC were used to extract the main set of features of the WfMS family. A Process Manager Pattern, developed within our group, was also used to exploit the WfMS domain. This is an architectural pattern for definition of Process-Centred Software Engineering Environment (PSEE) process managers.

The product line component architecture includes the following components:

- *GraphicalInterfaceMgr*: responsible for user interface management.
- *WorkflowArchitectureMgr*: supports the definition and maintenance of workflow architectures.

- **WorkflowMgr:** responsible for the instantiation and management of projects that are associated with a workflow.
- **WorkflowExecutionMgr:** responsible for the control and management of the process tasks supporting the interaction between workflow users and the environment, in particular supporting the selection of tasks to be executed.
- **TaskScheduler:** responsible for task and action control and management. It allows the interaction of the users to the WfMS.
- **AllocationResourceMgr:** responsible for resource allocation (actors, tools or material). In addition to the resource type and tool type, variation points include allocation policies.
- **ExternalApplicationMgr:** responsible for the management of external applications during the workflow definition and task execution.
- **ObjectMgr:** responsible for the mechanism for object management.
- **Interpreter:** responsible for the execution of a workflow script written in a process programming language.

Variation points at the design level were defined for each of these components. The skeleton of the architecture product has been incrementally implemented by several research works. In these works the same design process as the one of the product line is followed. Thus, they exploit variation points within each component and their impact on the overall architecture.

3.2 The Product Generation Process

The generation process is based on the Kobra approach. It consists of: requirement analysis, instantiation and packing. The activities of each phase were specified to make their artefacts and roles associated explicit. UML stereotypes were used to allow the identification and tracking of variations points. A case study was developed where a commercial tool was used to track the variation points from use cases to objects. It takes the decision model as a start and produces an updated design of the product with variation points resolved.

3.3 Product Line Architecture Evaluation

In order to evaluate the proposed architecture without delving into implementation details, we opted for an Architecture Definition Language (ADL). The Rapide ADL [5] was chosen to specify the proposed architecture, as it is a general purpose ADL for the modelling of component interfaces and their externally visible behaviour. In addition, there is a support environment that allows the definition of the architecture and its simulation. The simulation was

carried out based on the selection of relevant scenarios for WfMS. Sequence diagrams were drawn to represent the interaction of each specific scenario using components in the place of objects. The architecture was executed according to the scenarios to simulate the behaviour of the system.

3.4 The Variability Management Process

The variability management process consists of: elaboration of use case and feature models, variability identification, variability constraining, selection of variability implementation mechanisms and configuration analysis. The activities of the variability management process are related and synchronised with the product line activities. In addition, a data model was conceived to allow the monitoring and tracking of variability.

4. Summary

This paper presented the research that has been carried out within our group to develop a component-based product line for WfMS. Product line is still a recent approach, which has many open research issues. In particular, there is demand for experimental techniques that increases the confidence on its promises. A product line involves most of the software engineering problems. So, adequate experimental strategies should tackle many issues. We have dealt with simple case studies to evaluate the architecture but we feel that well-defined metrics are still required. The scale of the case studies also needs to be improved. We would like to speak of the product line quality by drawing results that include aspects such as the spectrum of products it can successfully generate. However, there are not thousands of case studies available to run, so an experimental strategy is required.

5. References

- [1] P. Clements, L. Northrop, Software product lines: practices and patterns. 1. Ed., Addison-Wesley, 2001. 608 p.
- [2] Workflow Management Coalition. Workflow Reference Model. Document number TC00-1003, January 1995.
- [3] D. F. D'Souza, A. C. Wills, Objects, Components and Frameworks with UML – The Catalysis Approach. Addison Wesley Publishing Company, 1999.
- [4] M. Morisio, G. H. Travassos, M. Stark, "Extending UML to Support Domain Analysis," in 2000 Proc. IEEE International Conference on Automated Software Engineering, pp. 321-324.
- [5] Computer Science Lab, DRAFT Guide to Rapide 1.0 - Language Reference Manuals, Rapide Design Team. Stanford University, 1997.